

Complex Support Vector Machines for Regression and Quaternary Classification

Pantelis Bouboulis

*Department of Informatics and Telecommunications,
University of Athens, Greece,*

PANBOUBOULIS@GMAIL.COM

Sergios Theodoridis

*Department of Informatics and Telecommunications,
University of Athens, Greece,*

STHEODOR@DI.UOA.GR

Charalampos Mavroforakis

*Department of Computer Science,
Data Management Lab,
Boston University,
Boston, MA 02215, USA.*

CMAV@BU.EDU

Editor:

Abstract

We present a support vector machines (SVM) rationale suitable for regression and quaternary classification problems that use complex data, exploiting the notions of widely linear estimation and pure complex kernels. The recently developed Wirtinger's calculus on complex RKHS is employed in order to compute the Lagrangian and derive the dual optimization problem. We prove that this approach is equivalent with solving two real SVM tasks exploiting a specific real kernel, which is induced by the chosen complex kernel.

Keywords: Support Vector Machines, Kernel methods, Widely linear estimation, complex data

1. Introduction

The support vector machines (SVM) framework has become a popular toolbox for addressing non-linear classification and regression tasks. The excellent performance of SVMs was firmly grounded in the context of statistical learning theory (or VC theory as it is also called, giving credit to Vapnik and Chervonenkis, who developed it), which ensures their fine generalization properties. Today, support vector classifiers are amongst the most efficient algorithms for treating a large number of real world applications. In the context of regression, this toolbox is usually known as Support Vector Regression (SVR).

In its original form, the SVM method is a nonlinear generalization of the *Generalized Portrait* algorithm, which has been developed in the former USSR in the sixties. The introduction of non-linearity was carried out via a computationally elegant way known today to the machine learning community as the *kernel trick* Scholkopf and Smola (2002). Usually, this trick is applied in a black-box rationale:

“Given an algorithm which is formulated in terms of dot products, one can construct an alternative algorithm by replacing each one of the dot products with a positive definite kernel κ .”

The successful application of the kernel trick in SVMs has sparked a new breed of techniques for addressing non linear tasks, the so called *kernel-based methods*. Currently, kernel-based algorithms are a popular tool employed in a variety of scientific domains, ranging from adaptive filtering Slavakis et al. (2013 (to appear); Müller et al. (2001) and image processing to biology and nuclear physics Scholkopf and Smola (2002); Theodoridis and Koutroumbas (2008); Vapnik (1999); Shawe-Taylor and Cristianini (2004); Liu et al. (2010); Kivinen et al. (2004); Engel et al. (2004); Slavakis et al. (2008); Liu et al. (2008); Slavakis et al. (2009); Mavroforakis et al. (2007); Theodoridis and Mavroforakis (2007); Bouboulis et al. (2010); Bouboulis and Theodoridis (2011); Scholkopf et al. (2004); Müller et al. (1997).

In kernel-based methods, the notion of the Reproducing Kernel Hilbert Space (RKHS) plays a significant role. The original data are transformed into a higher dimensional RKHS \mathcal{H} (possibly of infinite dimension) and linear tools are applied to the transformed data in the so called feature space \mathcal{H} . This is equivalent to solving a non-linear problem in the original space. Furthermore, inner products in \mathcal{H} can efficiently be computed via the specific kernel function κ associated to the RKHS \mathcal{H} , disregarding the actual structure of the space. Recently, this rationale has been generalized, so that the task simultaneously learns the so called kernel in some fashion, instead of selecting it a priori, in the context of Multiple Kernel Learning (MKL) Bach et al. (2004); Bach (2008); Gönen and Alpaydin (2011); Sonnenburg et al. (2006).

Although the theory of RKHS has been developed by the mathematicians for general complex spaces, most kernel-based methods employ real kernels. This is largely due to the fact that many of them originated as variants of the original SVM formulation, which was targeted to treat real data. However, in modern applications complex data arise frequently in areas as diverse as communications, biomedicine, radar, etc. The complex domain provides a convenient and elegant representation for such data, but also a natural way to preserve their characteristics and to handle transformations that need to be performed. Hence, the design of SVMs suitable for treating problems of complex and/or multidimensional outputs has attracted some attention in the machine learning community. Perhaps the most complete works, which attempt to generalize the SVM rationale in this fashion, are a) the Clifford SVMs Bayro-Corrochano and Arana-Daniel (2010) and b) the division algebraic SVR Shilton and Lai (2007); Shilton et al. (2010, 2012). In the Clifford SVMs, the authors use Clifford algebras to extend the SVMs framework to multidimensional outputs. Clifford algebras belong to a type of associative algebras, which are used in mathematics to generalize the complex numbers, quaternions and several other hypercomplex number systems. On the other hand, in the division algebraic SVR, division algebras are employed for the same purpose. These are algebras, closely related to the Clifford ones, where all non-zero elements have multiplicative inverses. In a nutshell, Clifford algebras are more general and they can be employed to create a general algebraic framework (i.e., addition and multiplication operations) in any type of vector spaces (e.g., \mathbb{R} , \mathbb{R}^2 , \mathbb{R}^3 , \dots), while the division algebras are only four: the real numbers, the complex numbers (\mathbb{R}^2), the quaternions (\mathbb{R}^4) and the octonions (\mathbb{R}^8). This is due to the fact that the need for inverses can only be satisfied in these four vector spaces. Although Clifford algebras are more general, their limitations

(e.g., the lack of inverses) make them a difficult tool to work with, compared to the division algebras. Another notable attempt that pursue similar goals is the multiregression SVMs of Sanchez-Fernandez et al. (2004), where the outputs are represented simply as vectors and an ϵ -insensitive loss function is adopted. Unfortunately this approach does not result to a well defined dual problem. In contrast to the more general case of hyper-complex outputs, where applications are limited Che Ujang et al. (2011), complex valued SVMs have been adopted by a number of authors for the beamforming problem (e.g., Ramon et al. (2005); Gaudes et al. (2007)), although restricted to the simple linear case.

It is important to emphasize that all the aforementioned efforts to apply the SVM rationale to complex and hypercomplex numbers are limited to the case of the output data. These methods consider a multidimensional output, which can be represented, for example, as a complex number or a quaternion, while the input data are real vectors. Moreover, they employ real valued kernels to model the input-output relationship, breaking it down to its multidimensional components. However, in this way many of the rich geometric characteristics of complex and hypercomplex spaces are lost. In this paper we propose a different approach to the problem of generalizing the SVM framework to complex spaces. Our modeling takes place directly into complex RKHS, which are generated by pure complex kernels, instead of real ones. In that fashion, the geometry of the complex space is preserved. To be inline with the current trend in complex signal processing, we employ the widely linear estimation process, which it has been shown to perform better than the standard linear one Bouboulis et al. (2012b); Adali and Li (2010); Novey and Adali (2008); Mandic and Goh (2009); Kuh and Mandic (2009). This means that we model the input-output relationship as a sum of two parts. The first is linear with respect to the input vector, while the second is linear with respect to its conjugate. Moreover, we show that in the case of complex SVMs, the widely linear approach is a necessity, as the alternative would lead to a significantly restricted model. In order to compute the gradients, which are required by the Karush-Kuhn-Tucker conditions and the dual, we employ the generalized Wirtinger Calculus introduced in Bouboulis and Theodoridis (2011).

As one of our major result, we prove that working in a complex RKHS \mathbb{H} , with a pure complex kernel $\kappa_{\mathbb{C}}$, is equivalent to solving two problems in a real RKHS \mathcal{H} , albeit with a specific real kernel $\kappa_{\mathbb{R}}$, which is induced by the complex $\kappa_{\mathbb{C}}$. It must be pointed out that these induced kernels are not trivial. For example, the exploitation of the complex gaussian kernel results to an induced kernel different from the standard real gaussian RBF. Our emphasis in this paper is to outline the theoretical development and to verify the validity of our results via some simulation examples. The paper is organized as follows. In Section 2 the main mathematical background regarding RKHS is outlined and the differences between real and complex RKHS's are highlighted. Section 3 describes the standard real SVM and SVR algorithms. The main contributions of the paper can be found in Sections 4 and 5, where the theory and the generalized complex algorithms are developed. The complex SVR developed there, is suitable for general complex valued function estimation problems defined on complex domains. The proposed complex SVM rationale, on the other hand, is suitable for Quaternary classification (i.e., four classes problem), in contrast to the binary classification carried out by the real SVM approach. Experiments are presented in Section 6. Finally, section 7 contains some concluding remarks.

2. Real and Complex RKHS

Throughout the paper, we will denote the set of all integers, real and complex numbers by \mathbb{N} , \mathbb{R} and \mathbb{C} respectively. The imaginary unit is denoted as \mathbf{i} . Vector or matrix valued quantities appear in boldfaced symbols. A RKHS Aronszajn (1950) is a Hilbert space \mathcal{H} over a field \mathbb{F} for which there exists a positive definite function $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{F}$ with the following two important properties: a) For every $x \in \mathcal{X}$, $\kappa(\cdot, x)$ belongs to \mathcal{H} and b) κ has the so called *reproducing property*, i.e., $f(x) = \langle f, \kappa(\cdot, x) \rangle_{\mathcal{H}}$, for all $f \in \mathcal{H}$, in particular $\kappa(x, y) = \langle \kappa(\cdot, y), \kappa(\cdot, x) \rangle_{\mathcal{H}}$. The map $\Phi : \mathcal{X} \rightarrow \mathcal{H} : \Phi(x) = \kappa(\cdot, x)$ is called the *feature map* of \mathcal{H} . Recall, that in the case of complex Hilbert spaces (i.e., $\mathbb{F} = \mathbb{C}$) the inner product is sesqui-linear (i.e., linear in one argument and antilinear in the other) and Hermitian. In the real case, the symmetry condition implies $\kappa(x, y) = \langle \kappa(\cdot, y), \kappa(\cdot, x) \rangle_{\mathcal{H}} = \langle \kappa(\cdot, x), \kappa(\cdot, y) \rangle_{\mathcal{H}}$. However, since in the complex case the inner product is Hermitian, the aforementioned condition is equivalent to $\kappa(x, y) = (\langle \kappa(\cdot, x), \kappa(\cdot, y) \rangle_{\mathcal{H}})^* = \kappa^*(y, x)$. In the following, we will denote by \mathbb{H} a complex RKHS and by \mathcal{H} a real one. Moreover, in order to distinguish the two cases, we will use the notations $\kappa_{\mathbb{R}}$ and $\Phi_{\mathbb{R}}$ to refer to a real kernel and its corresponding feature map, instead of the notation $\kappa_{\mathbb{C}}$, $\Phi_{\mathbb{C}}$, which is reserved for pure complex kernels.

Definitely, the most popular real kernel in the literature is the *Gaussian radial basis function*, i.e., $\kappa_{\mathbb{R}^{\nu}, t}(\mathbf{x}, \mathbf{y}) := \exp(-t \sum_{k=1}^{\nu} (x_k - y_k)^2)$, defined for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{\nu}$, where t is a free positive parameter. Many more can be found in Scholkopf and Smola (2002); Theodoridis and Koutroumbas (2008); Shawe-Taylor and Cristianini (2004); Bouboulis and Mavroforakis (2011). Correspondingly, an important complex kernel is the *complex Gaussian kernel*, which is defined as: $\kappa_{\mathbb{C}^{\nu}, t}(\mathbf{z}, \mathbf{w}) := \exp(-t \sum_{k=1}^{\nu} (z_k - w_k^*)^2)$, where $\mathbf{z}, \mathbf{w} \in \mathbb{C}^{\nu}$, z_k denotes the k -th component of the complex vector $\mathbf{z} \in \mathbb{C}^{\nu}$ and $\exp(\cdot)$ is the extended exponential function in the complex domain. Other examples include the Bergman and the Szego kernels Paulsen (2009).

Besides the complex RKHS produced by the associated complex kernels, such as the aforementioned ones, one may construct a complex RKHS as a cartesian product of a real RKHS with itself, in a fashion similar to the identification of the field of complex numbers, \mathbb{C} , to \mathbb{R}^2 . This technique is called *complexification* of a real RKHS and the respective Hilbert space is called *complexified* RKHS. Let $\mathcal{X} \subseteq \mathbb{R}^{\nu}$ and define the spaces $\mathcal{X}^2 \equiv \mathcal{X} \times \mathcal{X} \subseteq \mathbb{R}^{2\nu}$ and $\mathbb{X} = \{\mathbf{x} + \mathbf{i}\mathbf{y}, \mathbf{x}, \mathbf{y} \in \mathcal{X}\} \subseteq \mathbb{C}^{\nu}$, where the latter is equipped with a complex product structure. Let \mathcal{H} be a real RKHS associated with a real kernel $\kappa_{\mathbb{R}}$ defined on $\mathcal{X}^2 \times \mathcal{X}^2$ and let $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ be its corresponding inner product. Then, every $f \in \mathcal{H}$ can be regarded as a function defined on either \mathcal{X}^2 or \mathbb{X} , i.e., $f(\mathbf{z}) = f(\mathbf{x} + \mathbf{i}\mathbf{y}) = f(\mathbf{x}, \mathbf{y})$. Moreover, we define the cartesian product of \mathcal{H} with itself, i.e., $\mathcal{H}^2 = \mathcal{H} \times \mathcal{H}$. It is easy to verify that \mathcal{H}^2 is also a Hilbert Space with inner product

$$\langle \mathbf{f}, \mathbf{g} \rangle_{\mathcal{H}^2} = \langle f^r, g^r \rangle_{\mathcal{H}} + \langle f^i, g^i \rangle_{\mathcal{H}}, \quad (1)$$

for $\mathbf{f} = (f^r, f^i)$, $\mathbf{g} = (g^r, g^i)$. Our objective is to enrich \mathcal{H}^2 with a complex structure. To this end, we define the space $\mathbb{H} = \{f = f^r + \mathbf{i}f^i; f^r, f^i \in \mathcal{H}\}$ equipped with the complex inner product:

$$\langle f, g \rangle_{\mathbb{H}} = \langle f^r, g^r \rangle_{\mathcal{H}} + \langle f^i, g^i \rangle_{\mathcal{H}} + \mathbf{i}(\langle f^i, g^r \rangle_{\mathcal{H}} - \langle f^r, g^i \rangle_{\mathcal{H}}), \quad (2)$$

for $f = f^r + \mathbf{i}f^i$, $g = g^r + \mathbf{i}g^i$. It is not difficult to verify that the complexified space \mathbb{H} is a complex RKHS with kernel κ Paulsen (2009). We call \mathbb{H} the complexification of \mathcal{H} . It

can readily be seen, that, although \mathbb{H} is a complex RKHS, its respective kernel is real (i.e., its imaginary part is equal to zero). To complete the presentation of the complexification procedure, we need a technique to implicitly map the data samples from the complex input space to the complexified RKHS \mathbb{H} . This can be done using the simple rule:

$$\bar{\Phi}_{\mathbb{C}}(\mathbf{z}) = \bar{\Phi}_{\mathbb{C}}(\mathbf{x} + \mathbf{i}\mathbf{y}) = \bar{\Phi}_{\mathbb{C}}(\mathbf{x}, \mathbf{y}) = \Phi_{\mathbb{R}}(\mathbf{x}, \mathbf{y}) + \mathbf{i}\Phi_{\mathbb{R}}(\mathbf{x}, \mathbf{y}), \quad (3)$$

where $\Phi_{\mathbb{R}}$ is the feature map of the real reproducing kernel $\kappa_{\mathbb{R}}$, i.e., $\Phi_{\mathbb{R}}(\mathbf{x}, \mathbf{y}) = \kappa_{\mathbb{R}}(\cdot, (\mathbf{x}, \mathbf{y}))$ and $\mathbf{z} = \mathbf{x} + \mathbf{i}\mathbf{y}$. As a consequence, observe that:

$$\langle \bar{\Phi}_{\mathbb{C}}(\mathbf{z}), \bar{\Phi}_{\mathbb{C}}(\mathbf{z}') \rangle_{\mathbb{H}} = 2\langle \Phi_{\mathbb{R}}(\mathbf{x}, \mathbf{y}), \Phi_{\mathbb{R}}(\mathbf{x}', \mathbf{y}') \rangle_{\mathcal{H}} = 2\kappa_{\mathbb{R}}((\mathbf{x}', \mathbf{y}'), (\mathbf{x}, \mathbf{y})).$$

We have to emphasize that a complex RKHS \mathbb{H} (whether it is constructed through the complexification procedure, or it is produced by a complex kernel) can, always, be represented as a cartesian product of a Hilbert space with itself, i.e., we can, always, identify \mathbb{H} with a *doubled real space* \mathcal{H}^2 . Furthermore, the complex inner product of \mathbb{H} can always be related to the real inner product of \mathcal{H} as in (2).

In order to compute the gradients of real valued cost functions, which are defined on complex domains, we adopt the rationale of Wirtinger's calculus Wirtinger (1927). This was brought into light recently Adali and Li (2010); Novey and Adali (2008); Li (2008), as a means to compute, in an efficient and elegant way, gradients of real valued cost functions that are defined on complex domains (\mathbb{C}^{ν}), in the context of widely linear processing Mandic and Goh (2009); Picinbono and Chevalier (1995). It is based on simple rules and principles, which bear a great resemblance to the rules of the standard complex derivative, and it greatly simplifies the calculations of the respective derivatives. The difficulty with real valued cost functions is that they do not obey the Cauchy-Riemann conditions and are not differentiable in the complex domain. The alternative to Wirtinger's calculus would be to consider the complex variables as pairs of two real ones and employ the common real partial derivatives. However, this approach, usually, is more time consuming and leads to more cumbersome expressions. In Bouboulis and Theodoridis (2011), the notion of Wirtinger's calculus was extended to general complex Hilbert spaces, providing the tool to compute the gradients that are needed to develop kernel-based algorithms for treating complex data. In Bouboulis et al. (2012a) the notion of Wirtinger calculus was extended to include subgradients in RKHS

3. Real valued Support Vector Machines

In this section we briefly describe the popular SVM rational for real-valued data.

3.1 Support Vector Machines for Classification

Suppose we are given training data, which belong to two separate classes C_+, C_- and have the form $\{(\mathbf{x}_n, d_n); n = 1, \dots, N\} \subset \mathcal{X} \times \{\pm 1\}$. If $d_n = +1$, then the n -th sample belongs to C_+ , while if $d_n = -1$, then the n -th sample belongs to C_- . Consider the real RKHS \mathcal{H} with respective kernel $\kappa_{\mathbb{R}}$. We transform the input data from \mathcal{X} to \mathcal{H} , via the feature map $\Phi_{\mathbb{R}}$. The goal of the SVM task is to estimate the maximum margin hyperplane, that separates the points of the two classes as best as possible. As any hyperplane of \mathcal{H}

has the form $\langle f, w \rangle_{\mathcal{H}} + c = 0$, $f \in \mathcal{H}$, for some parameters $w \in \mathcal{H}$, $c \in \mathbb{R}$, the SVM task can be casted as Scholkopf and Smola (2002); Shawe-Taylor and Cristianini (2004); Theodoridis and Koutroumbas (2008):

$$\begin{aligned} & \underset{w \in \mathcal{H}, c \in \mathbb{R}}{\text{minimize}} && \frac{1}{2} \|w\|_{\mathcal{H}}^2 + \frac{C}{N} \sum_{n=1}^N \xi_n \\ & \text{subject to} && \begin{cases} d_n (\langle \Phi_{\mathbb{R}}(\mathbf{x}_n), w \rangle_{\mathcal{H}} + c) \geq 1 - \xi_n \\ \xi_n \geq 0 \end{cases} \\ & && \text{for } n = 1, \dots, N, \end{aligned} \quad (4)$$

for some $C > 0$. This is a constant that determines the trade-off between the two conflicting goals of the SVM task: maximizing the margin (i.e., $2/\|w\|^2$) and minimizing the training error (i.e., $\sum_{n=1}^N \xi_n$). The optimization task (4) is often called as the C-SVM classifier, to distinguish this case from other SVM formulations, such as the ν -SVM Theodoridis and Koutroumbas (2008).

Introducing the Lagrangian and exploiting the KKT conditions we find that the dual problem is casted as:

$$\begin{aligned} & \underset{\mathbf{a} \in \mathbb{R}^N}{\text{maximize}} && \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n,m=1}^N a_n a_m d_n d_m \kappa_{\mathbb{R}}(\mathbf{x}_m, \mathbf{x}_n) \\ & \text{subject to} && \sum_{n=1}^N a_n d_n = 0 \text{ and } a_n \in [0, C/N]. \end{aligned} \quad (5)$$

Furthermore, the solution can be shown to have an expansion:

$$w = \sum_{n=1}^N a_n d_n \kappa_{\mathbb{R}}(\cdot, \mathbf{x}_n),$$

while the threshold c can be computed by averaging

$$c = d_m - \sum_{n=1}^N a_n d_n \kappa_{\mathbb{R}}(\mathbf{x}_m, \mathbf{x}_n),$$

over all points with $0 < a_m < C$, for $m = 1, \dots, N$.

3.2 Support Vector Regression

In a more general setting, the outputs d_n may take several values or they may be real numbers. In the latter case, we are trying to estimate an input-output relationship between \mathbf{x}_n and d_n . The SVM rationale can be modified to accommodate this set-up as follows. Suppose we are given training data of the form $\{(\mathbf{x}_n, d_n); n = 1, \dots, N\} \subset \mathcal{X} \times \mathbb{R}$, where $\mathcal{X} = \mathbb{R}^p$ denotes the space of input patterns. Furthermore, let \mathcal{H} be a real RKHS with kernel $\kappa_{\mathbb{R}}$. We transform the input data from \mathcal{X} to \mathcal{H} , via the feature map $\Phi_{\mathbb{R}}$, to obtain the data $\{(\Phi_{\mathbb{R}}(\mathbf{x}_n), d_n); n = 1, \dots, N\}$. In support vector regression, the goal is to find an affine function $T : \mathcal{H} \rightarrow \mathbb{R} : T(f) = \langle f, w \rangle_{\mathcal{H}} + c$, for some $w \in \mathcal{H}$, $c \in \mathbb{R}$, which is as

flat as possible and has at most ϵ deviation from the actually obtained values d_n , for all $n = 1, \dots, N$. Observe that at the training points $\Phi_{\mathbb{R}}(\mathbf{x}_n)$, T takes the values $T(\Phi_{\mathbb{R}}(\mathbf{x}_n))$. Thus, this is equivalent with finding a non-linear function g defined on \mathcal{X} such that

$$g(\mathbf{x}) = T \circ \Phi_{\mathbb{R}}(\mathbf{x}) = \langle \Phi_{\mathbb{R}}(\mathbf{x}), w \rangle_{\mathcal{H}} + c, \quad (6)$$

for some $w \in \mathcal{H}$, $c \in \mathbb{R}$, which satisfies the aforementioned properties. The usual formulation of this problem as an optimization task is the following:

$$\begin{aligned} & \underset{w \in \mathcal{H}, c \in \mathbb{R}}{\text{minimize}} && \frac{1}{2} \|w\|_{\mathcal{H}}^2 + \frac{C}{N} \sum_{n=1}^N (\xi_n + \hat{\xi}_n) \\ & \text{subject to} && \begin{cases} \langle \Phi_{\mathbb{R}}(\mathbf{x}_n), w \rangle_{\mathcal{H}} + c - d_n \leq \epsilon + \xi_n \\ d_n - \langle \Phi_{\mathbb{R}}(\mathbf{x}_n), w \rangle_{\mathcal{H}} - c \leq \epsilon + \hat{\xi}_n \\ \xi_n, \hat{\xi}_n \geq 0 \end{cases} \end{aligned} \quad (7)$$

for $n = 1, \dots, N$. The constant C determines a tradeoff between the tolerance of the estimation (i.e., how many larger than ϵ deviations are tolerated) and the flatness of the solution (i.e., T). This corresponds to the so called ϵ -insensitive loss function $|\xi|_{\epsilon} = \max\{0, |\xi| - \epsilon\}$, Vapnik (1999); Theodoridis and Koutroumbas (2008).

To solve (7), one considers the dual problem derived by the Lagrangian:

$$\begin{aligned} & \underset{\mathbf{a}, \hat{\mathbf{a}}}{\text{maximize}} && \begin{cases} -\frac{1}{2} \sum_{n,m=1}^N (\hat{a}_n - a_n)(\hat{a}_m - a_m) \kappa(x_n, x_m) \\ -\epsilon \sum_{n=1}^N (\hat{a}_n + a_n) + \sum_{n=1}^N d_n (\hat{a}_n - a_n) \end{cases} \\ & \text{subject to} && \sum_{n=1}^N (\hat{a}_n - a_n) = 0 \text{ and } a_n, \hat{a}_n \in [0, C/N]. \end{aligned} \quad (8)$$

Note that a_n and \hat{a}_n are the Lagrange multipliers corresponding to the first two inequalities of problem (15), for $n = 1, 2, \dots, N$. Exploiting the saddle point conditions, it can be proved that $w = \sum_{n=1}^N (a_n - \hat{a}_n) \Phi(\mathbf{x}_n)$ and thus the solution becomes

$$f(\mathbf{x}) = \sum_{n=1}^N (\hat{a}_n - a_n) \kappa_{\mathbb{R}}(\mathbf{x}_n, \mathbf{x}) + c. \quad (9)$$

Furthermore, exploiting the Karush-Kuhn-Tucker (KKT) conditions one may compute the parameter c .

Several algorithms have been proposed for solving the SVM and SVR tasks, amongst which are Platt's celebrated Sequential Minimal Optimization (SMO) algorithm Platt (1998), interior point methods Vanderbei (1994), geometric algorithms Mavroforakis and Theodoridis (2006); Mavroforakis et al. (2007); Lázaró (2011) and methods suitable for large scale problems Collobert and Bengio (2001). A more detailed description of the SVR machinery can be found in Smola and Schölkopf (1998).

4. Complex Support Vector Regression

We begin the treatment of the complex case with the complex SVR rationale, as this is a direct generalization of the real SVR. Suppose we are given training data of the form $\{(\mathbf{z}_n, d_n); n = 1, \dots, N\} \subset \mathcal{X} \times \mathbb{C}$, where $\mathcal{X} = \mathbb{C}^\nu$ denotes the space of input patterns. As \mathbf{z}_n is complex, we denote by \mathbf{x}_n its real part and by \mathbf{y}_n its imaginary part respectively, i.e., $\mathbf{z}_n = \mathbf{x}_n + \mathbf{i}\mathbf{y}_n$, $n = 1, \dots, N$. Similarly, we denote by d_n^r and d_n^i the real and the imaginary part of d_n , i.e., $d_n = d_n^r + \mathbf{i}d_n^i$, $n = 1, \dots, N$.

4.1 Dual Channel SVR

A straightforward approach for addressing this problem (as well as any problem related with complex data) is by considering two different problems in the real domain. This technique is usually referred to as the *Dual Real Channel (DRC) approach*. That is, the training data are split into two sets $\{((\mathbf{x}_n, \mathbf{y}_n)^T, d_n^r); n = 1, \dots, N\} \subset \mathbb{R}^{2\nu} \times \mathbb{R}$ and $\{((\mathbf{x}_n, \mathbf{y}_n)^T, d_n^i); n = 1, \dots, N\} \subset \mathbb{R}^{2\nu} \times \mathbb{R}$, and perform support vector regression to each set of data using a real kernel $\kappa_{\mathbb{R}}$ and its corresponding RKHS. We will show in the following sections that the DRC approach is equivalent to the complexification procedure Bouboulis and Theodoridis (2011) described in section 2. The latter, however, often provides a context that enables us to work with complex data in a more compact form, as one may employ Wirtinger's Calculus to compute the respective gradients and develop algorithms directly in complex form Bouboulis and Theodoridis (2011).

In contrast to the complexification procedure, we emphasize that the pure complex approach (where one directly exploits a complex RKHS) considered in the next section is quite different from the DRC rationale. We will develop a framework for solving such a problem on the complex domain employing pure complex kernels, instead of real ones. Nevertheless, we will show that using complex kernels for SVR is equivalent with solving two real problems using a real kernel. This kernel, however, is induced by the selected complex kernel and *it is not one of the standard kernels* appearing in machine learning literature. For example, the use of the complex Gaussian kernel induces a real kernel, which is not the standard real Gaussian RBF (see figure 1). As it has already been demonstrated in Bouboulis et al. (2012a,b), although in a different context than the one we use here, the DRC approach and the pure complex approaches give, in general, different results. Depending on the case, the pure complex approach might show increased performance over the DRC approach and vice versa.

4.2 Pure Complex SVR

Prior to the development of the generalized complex SVR rationale, we investigate some significant properties of the complex kernels. In the following, we assume that \mathbb{H} is a complex RKHS with kernel $\kappa_{\mathbb{C}}$. We can decompose $\kappa_{\mathbb{C}}$ into its real and imaginary parts, i.e., $\kappa_{\mathbb{C}}(\mathbf{z}, \mathbf{z}') = \kappa_{\mathbb{C}}^r(\mathbf{z}, \mathbf{z}') + \mathbf{i}\kappa_{\mathbb{C}}^i(\mathbf{z}, \mathbf{z}')$, where $\kappa_{\mathbb{C}}^r(\mathbf{z}, \mathbf{z}'), \kappa_{\mathbb{C}}^i(\mathbf{z}, \mathbf{z}') \in \mathbb{R}$. As any complex kernel is Hermitian (see section 2), we have that $\kappa_{\mathbb{C}}^*(\mathbf{z}, \mathbf{z}') = \kappa_{\mathbb{C}}(\mathbf{z}', \mathbf{z})$ and hence we take

$$\kappa_{\mathbb{C}}^r(\mathbf{z}, \mathbf{z}') = \kappa_{\mathbb{C}}^r(\mathbf{z}', \mathbf{z}), \quad (10)$$

$$\kappa_{\mathbb{C}}^i(\mathbf{z}, \mathbf{z}') = -\kappa_{\mathbb{C}}^i(\mathbf{z}', \mathbf{z}). \quad (11)$$

Lemma 1 *The imaginary part of any complex kernel, $\kappa_{\mathbb{C}}$, satisfies:*

$$\sum_{n,m=1}^N c_n c_m \kappa_{\mathbb{C}}^i(\mathbf{z}_n, \mathbf{z}_m) = 0, \quad (12)$$

for any $N > 0$ and any selection of $c_1, \dots, c_N \in \mathbb{C}$ and $\mathbf{z}_1, \dots, \mathbf{z}_N \in \mathcal{X}$.

Proof Exploiting equation (11) and rearranging the indices of the summation we get:

$$\sum_{n,m=1}^N c_n c_m \kappa_{\mathbb{C}}^i(\mathbf{z}_n, \mathbf{z}_m) = - \sum_{n,m=1}^N c_n c_m \kappa_{\mathbb{C}}^i(\mathbf{z}_m, \mathbf{z}_n) = - \sum_{m,n=1}^N c_m c_n \kappa_{\mathbb{C}}^i(\mathbf{z}_n, \mathbf{z}_m).$$

Hence, $2 \sum_{n,m=1}^N c_n c_m \kappa_{\mathbb{C}}^i(\mathbf{z}_n, \mathbf{z}_m) = 0$ and the result follows immediately. \blacksquare

Lemma 2 *If $\kappa_{\mathbb{C}}(\mathbf{z}, \mathbf{z}')$ is a complex kernel defined on $\mathbb{C}^\nu \times \mathbb{C}^\nu$, then its real part, i.e.,*

$$\kappa_{\mathbb{C}}^r \left(\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}, \begin{pmatrix} \mathbf{x}' \\ \mathbf{y}' \end{pmatrix} \right) = \text{Re}(\kappa_{\mathbb{C}}(\mathbf{z}, \mathbf{z}')), \quad (13)$$

where $\mathbf{z} = \mathbf{x} + \mathbf{i}\mathbf{y}$, $\mathbf{z}' = \mathbf{x}' + \mathbf{i}\mathbf{y}'$, is a real kernel defined on $\mathbb{R}^{2\nu} \times \mathbb{R}^{2\nu}$. We call this kernel the induced real kernel of $\kappa_{\mathbb{C}}$.

Proof As relation (10) implies, $\kappa_{\mathbb{C}}^r$ is symmetric. Moreover, let $N > 0$, $\alpha_1, \dots, \alpha_N \in \mathbb{R}$ and $\mathbf{z}_1, \dots, \mathbf{z}_N \in \mathcal{X}$. As $\kappa_{\mathbb{C}}$ is positive definite, we have that

$$\sum_{n,m=1}^N \alpha_n \alpha_m \kappa_{\mathbb{C}}(\mathbf{z}_n, \mathbf{z}_m) \geq 0.$$

However, splitting $\kappa_{\mathbb{C}}$ to its real and imaginary parts and exploiting Lemma 1, we take

$$\begin{aligned} \sum_{n,m=1}^N \alpha_n \alpha_m \kappa_{\mathbb{C}}(\mathbf{z}_n, \mathbf{z}_m) &= \sum_{n,m=1}^N \alpha_n \alpha_m \kappa_{\mathbb{C}}^r(\mathbf{z}_n, \mathbf{z}_m) + \mathbf{i} \sum_{n,m=1}^N \alpha_n \alpha_m \kappa_{\mathbb{C}}^i(\mathbf{z}_n, \mathbf{z}_m) \\ &= \sum_{n,m=1}^N \alpha_n \alpha_m \kappa_{\mathbb{C}}^r(\mathbf{z}_n, \mathbf{z}_m). \end{aligned}$$

Hence, $\sum_{n,m=1}^N \alpha_n \alpha_m \kappa_{\mathbb{C}}^r(\mathbf{z}_n, \mathbf{z}_m) \geq 0$. As a last step, recall that $\kappa_{\mathbb{C}}^r$ may be regarded as defined either on $\mathbb{C}^\nu \times \mathbb{C}^\nu$ or $\mathbb{R}^{2\nu} \times \mathbb{R}^{2\nu}$. This leads to

$$\sum_{n,m=1}^N \alpha_n \alpha_m \kappa_{\mathbb{C}}^r \left(\begin{pmatrix} \mathbf{x}_n \\ \mathbf{y}_n \end{pmatrix}, \begin{pmatrix} \mathbf{x}_m \\ \mathbf{y}_m \end{pmatrix} \right) \geq 0.$$

We conclude that $\kappa_{\mathbb{C}}^r$ is a positive definite kernel on $\mathbb{R}^{2\nu} \times \mathbb{R}^{2\nu}$. ■

At this point we are ready to present the SVR rationale in complex RKHS. We transform the input data from \mathcal{X} to \mathbb{H} , via the feature map $\Phi_{\mathbb{C}}$, to obtain the data $\{(\Phi_{\mathbb{C}}(z_n), d_n); n = 1, \dots, N\}$. In analogy with the real case and extending the principles of widely linear estimation to complex support vector regression, the goal is to find a function $T : \mathbb{H} \rightarrow \mathbb{C} : T(f) = \langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c$, for some $u, v \in \mathbb{H}$, $c \in \mathbb{C}$, which is as flat as possible and has at most ϵ deviation from both the real and imaginary parts of the actually obtained values d_n , for all $n = 1, \dots, N$. We emphasize that we employ the widely linear estimation function $S_1 : \mathbb{H} \rightarrow \mathbb{C} : S_1(f) = \langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}}$ instead of the usual complex linear function $S_2 : \mathbb{H} \rightarrow \mathbb{C} : S_2(f) = \langle f, w \rangle_{\mathbb{H}}$ following the ideas of Picinbono and Chevalier (1995), which are becoming popular in complex signal processing Took and Mandic (2010); Chevalier and Pipon (2006); Jeon et al. (2006); Cacciapuoti et al. (2008) and have been generalized for the case of complex RKHS in Bouboulis et al. (2012a). It has been established Picinbono (1994); Picinbono and Bondon (1997), that the widely linear estimation functions are able to capture the second order statistical characteristics of the input data, which are necessary if non-circular¹ input sources are considered. Furthermore, as it has been shown in Bouboulis et al. (2012b), the exploitation of the traditional complex linear function excludes a significant percentage of linear functions from being considered in the estimation process. The correct and natural linear estimation in complex spaces is the widely linear one.

Observe that at the training points $\Phi_{\mathbb{C}}(z_n)$, T takes the values $T(\Phi_{\mathbb{C}}(z_n))$. Following similar arguments as with the real case, this is equivalent with finding a complex non-linear function g defined on \mathcal{X} such that

$$g(z) = T \circ \Phi_{\mathbb{C}}(z) = \langle \Phi_{\mathbb{C}}(z), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}^*(z), v \rangle_{\mathbb{H}} + c, \quad (14)$$

for some $w, v \in \mathbb{H}$, $c \in \mathbb{C}$, which satisfies the aforementioned properties. We formulate the complex support vector regression task as follows:

$$\begin{aligned} \min_{w, v, c} \quad & \frac{1}{2} \|w\|_{\mathbb{H}}^2 + \frac{1}{2} \|v\|_{\mathbb{H}}^2 + \frac{C}{N} \sum_{n=1}^N (\xi_n^r + \hat{\xi}_n^r + \xi_n^i + \hat{\xi}_n^i) \\ \text{s. t.} \quad & \begin{cases} \text{Re}(\langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}^*(z_n), v \rangle_{\mathbb{H}} + c - d_n) \leq \epsilon + \xi_n^r \\ \text{Re}(d_n - \langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} - \langle \Phi_{\mathbb{C}}^*(z_n), v \rangle_{\mathbb{H}} - c) \leq \epsilon + \hat{\xi}_n^r \\ \text{Im}(\langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}^*(z_n), v \rangle_{\mathbb{H}} + c - d_n) \leq \epsilon + \xi_n^i \\ \text{Im}(d_n - \langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} - \langle \Phi_{\mathbb{C}}^*(z_n), v \rangle_{\mathbb{H}} - c) \leq \epsilon + \hat{\xi}_n^i \\ \xi_n^r, \hat{\xi}_n^r, \xi_n^i, \hat{\xi}_n^i \geq 0 \end{cases} \end{aligned} \quad (15)$$

1. Note that the issue of circularity has become quite popular recently in the context of complex adaptive filtering. Circularity is intimately related to rotation in the geometric sense. A complex random variable Z is called circular, if for any angle ϕ both Z and $Ze^{i\phi}$ (i.e., the rotation of Z by angle ϕ) follow the same probability distribution Mandic and Goh (2009).

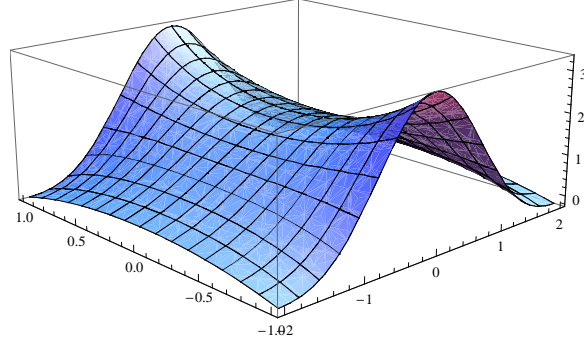


Figure 1: The element $\kappa_{\mathbb{C}}^r(\cdot, (0, 0)^T)$ of the induced real feature space of the complex Gaussian kernel.

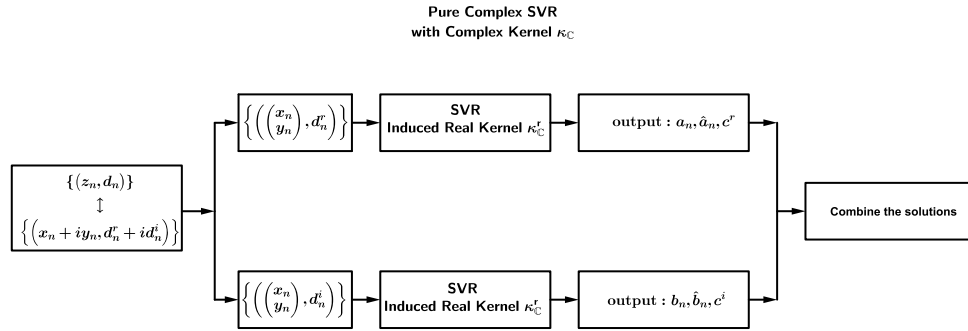


Figure 2: Pure Complex Support Vector Regression. The difference with the dual channel approach is due to the incorporation of the induced real kernel $\kappa_{\mathbb{C}}^r$, which depends on the selection of the complex kernel $\kappa_{\mathbb{C}}$. In this context one exploits the complex structure of the space, which is lost in the dual channel approach.

To solve (15), we derive the Lagrangian and the KKT conditions to obtain the dual problem. Thus we take:

$$\begin{aligned}
\mathcal{L} = & \frac{1}{2}\|w\|^2 + \frac{1}{2}\|v\|^2 + \frac{C}{N} \sum_{n=1}^N (\xi_n^r + \hat{\xi}_n^r + \xi_n^i + \hat{\xi}_n^i) \\
& + \sum_{n=1}^N a_n (\text{Re}(\langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}(z_n), v \rangle_{\mathbb{H}} + c - d_n) - \epsilon - \xi_n^r) \\
& + \sum_{n=1}^N \hat{a}_n (\text{Re}(d_n - \langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} - \langle \Phi_{\mathbb{C}}^*(z_n), v \rangle_{\mathbb{H}} - c) - \epsilon - \hat{\xi}_n^r) \\
& + \sum_{n=1}^N b_n (\text{Im}(\langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}(z_n), v \rangle_{\mathbb{H}} + c - d_n) - \epsilon - \xi_n^i) \\
& + \sum_{n=1}^N \hat{b}_n (\text{Im}(d_n - \langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} - \langle \Phi_{\mathbb{C}}^*(z_n), v \rangle_{\mathbb{H}} - c) - \epsilon + \hat{\xi}_n^i) \\
& - \sum_{n=1}^N \eta_n \xi_n^r - \sum_{n=1}^N \hat{\eta}_n \hat{\xi}_n^r - \sum_{n=1}^N \theta_n \xi_n^i - \sum_{n=1}^N \hat{\theta}_n \hat{\xi}_n^i,
\end{aligned} \tag{16}$$

where $a_n, \hat{a}_n, b_n, \hat{b}_n, \eta_n, \hat{\eta}_n, \theta_n, \hat{\theta}_n$ are the Lagrange multipliers. To exploit the saddle point conditions, we employ the rules of Wirtinger's Calculus for the complex variables on complex RKHS's as described in Bouboulis and Theodoridis (2011) and deduce that

$$\frac{\partial \mathcal{L}}{\partial w^*} = \frac{1}{2}w + \frac{1}{2} \sum_{n=1}^N a_n \Phi_{\mathbb{C}}(z_n) - \frac{1}{2} \sum_{n=1}^N \hat{a}_n \Phi_{\mathbb{C}}(z_n) - \frac{\mathbf{i}}{2} \sum_{n=1}^N b_n \Phi_{\mathbb{C}}(z_n) + \frac{\mathbf{i}}{2} \sum_{n=1}^N \hat{b}_n \Phi_{\mathbb{C}}(z_n),$$

$$\frac{\partial \mathcal{L}}{\partial v^*} = \frac{1}{2}v + \frac{1}{2} \sum_{n=1}^N a_n \Phi_{\mathbb{C}}^*(z_n) - \frac{1}{2} \sum_{n=1}^N \hat{a}_n \Phi_{\mathbb{C}}^*(z_n) - \frac{\mathbf{i}}{2} \sum_{n=1}^N b_n \Phi_{\mathbb{C}}^*(z_n) + \frac{\mathbf{i}}{2} \sum_{n=1}^N \hat{b}_n \Phi_{\mathbb{C}}^*(z_n),$$

$$\frac{\partial \mathcal{L}}{\partial c^*} = \frac{1}{2} \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \hat{a}_n + \frac{\mathbf{i}}{2} \sum_{n=1}^N b_n - \frac{\mathbf{i}}{2} \sum_{n=1}^N \hat{b}_n.$$

For the real variables we compute the gradients in the traditional way:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \xi_n^r} &= \frac{C}{N} - a_n - \eta_n, & \frac{\partial \mathcal{L}}{\partial \hat{\xi}_n^r} &= \frac{C}{N} - \hat{a}_n - \hat{\eta}_n, \\
\frac{\partial \mathcal{L}}{\partial \xi_n^i} &= \frac{C}{N} - b_n - \theta_n, & \frac{\partial \mathcal{L}}{\partial \hat{\xi}_n^i} &= \frac{C}{N} - \hat{b}_n - \hat{\theta}_n.
\end{aligned}$$

for all $n = 1, \dots, N$.

As all gradients have to vanish for the saddle point conditions, we finally take that

$$w = \sum_{n=1}^N (\hat{a}_n - a_n) \Phi_{\mathbb{C}}(z_n) - \mathbf{i} \sum_{n=1}^N (\hat{b}_n - b_n) \Phi_{\mathbb{C}}(z_n), \tag{17}$$

$$v = \sum_{n=1}^N (\hat{a}_n - a_n) \Phi_{\mathbb{C}}^*(z_n) - \mathbf{i} \sum_{n=1}^N (\hat{b}_n - b_n) \Phi_{\mathbb{C}}^*(z_n), \tag{18}$$

$$\sum_{n=1}^N (\hat{a}_n - a_n) = \sum_{n=1}^N (\hat{b}_n - b_n) = 0, \quad (19)$$

$$\begin{aligned} \eta_n &= \frac{C}{N} - a_n, & \hat{\eta}_n &= \frac{C}{N} - \hat{a}_n, \\ \theta_n &= \frac{C}{N} - b_n, & \hat{\theta}_n &= \frac{C}{N} - \hat{b}_n, \end{aligned} \quad (20)$$

for $n = 1, \dots, N$.

To compute $\|w\|_{\mathbb{H}}^2 = \langle w, w \rangle_{\mathbb{H}}$, we apply equation (17), Lemma 1, the reproducing property of \mathbb{H} , i.e., $\langle \Phi(\mathbf{z}_n), \Phi(\mathbf{z}_m) \rangle_{\mathbb{H}} = \kappa_{\mathbb{C}}(\mathbf{z}_m, \mathbf{z}_n)$, and the sesqui-linear property of the inner product of \mathbb{H} to obtain that:

$$\begin{aligned} \|w\|_{\mathbb{H}}^2 &= \sum_{n,m=1}^N (\hat{a}_n - a_n)(\hat{a}_m - a_m) \kappa_{\mathbb{C}}^r(\mathbf{z}_m, \mathbf{z}_n) + \sum_{n,m=1}^N (\hat{b}_n - b_n)(\hat{b}_m - b_m) \kappa_{\mathbb{C}}^r(\mathbf{z}_m, \mathbf{z}_n) \\ &\quad + 2 \sum_{n,m=1}^N (\hat{a}_n - a_n)(\hat{b}_m - b_m) \kappa_{\mathbb{C}}^i(\mathbf{z}_m, \mathbf{z}_n). \end{aligned}$$

Similarly, we have

$$\begin{aligned} \|v\|_{\mathbb{H}}^2 &= \sum_{n,m=1}^N (\hat{a}_n - a_n)(\hat{a}_m - a_m) \kappa_{\mathbb{C}}^r(\mathbf{z}_m, \mathbf{z}_n) + \sum_{n,m=1}^N (\hat{b}_n - b_n)(\hat{b}_m - b_m) \kappa_{\mathbb{C}}^r(\mathbf{z}_m, \mathbf{z}_n) \\ &\quad - 2 \sum_{n,m=1}^N (\hat{a}_n - a_n)(\hat{b}_m - b_m) \kappa_{\mathbb{C}}^i(\mathbf{z}_m, \mathbf{z}_n), \end{aligned}$$

and

$$\frac{\langle \Phi_{\mathbb{C}}(\mathbf{z}_n), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}^*(\mathbf{z}_n), v \rangle_{\mathbb{H}}}{2} = \sum_{m=1}^N (\hat{a}_m - a_m) \kappa_{\mathbb{C}}^r(\mathbf{z}_m, \mathbf{z}_n) + \mathfrak{i} \sum_{m=1}^N (\hat{b}_m - b_m) \kappa_{\mathbb{C}}^r(\mathbf{z}_m, \mathbf{z}_n).$$

Eliminating $\eta_n, \hat{\eta}_n, \theta_n, \hat{\theta}_n$ via (20) and w, v via the aforementioned relations, we obtain the final form of the Lagrangian:

$$\begin{aligned} \mathcal{L} = & - \sum_{n,m=1}^N (\hat{a}_n - a_n)(\hat{a}_m - a_m) \kappa_{\mathbb{C}}^r(\mathbf{z}_m, \mathbf{z}_n) - \sum_{n,m=1}^N (\hat{b}_n - b_n)(\hat{b}_m - b_m) \kappa_{\mathbb{C}}^r(\mathbf{z}_m, \mathbf{z}_n) \\ & - \epsilon \sum_{n=1}^N (a_n + \hat{a}_n + b_n + \hat{b}_n) + \sum_{n=1}^N d_n^r (\hat{a}_n - a_n) + \sum_{n=1}^N d_n^i (\hat{b}_n - b_n), \end{aligned} \quad (21)$$

where d_n^r, d_n^i are the real and imaginary parts of the output d_n , $n = 1, \dots, N$. This means that we can split the dual problem into two separate maximization tasks:

$$\begin{aligned} & \underset{\mathbf{a}, \hat{\mathbf{a}}}{\text{maximize}} \quad \begin{cases} - \sum_{n,m=1}^N (\hat{a}_n - a_n)(\hat{a}_m - a_m) \kappa_{\mathbb{C}}^r(\mathbf{z}_m, \mathbf{z}_n) \\ - \epsilon \sum_{n=1}^N (\hat{a}_n + a_n) + \sum_{n=1}^N d_n^r (\hat{a}_n - a_n) \end{cases} \\ & \text{subject to} \quad \sum_{n=1}^N (\hat{a}_n - a_n) = 0 \text{ and } a_n, \hat{a}_n \in [0, C/N], \end{aligned} \quad (22a)$$

and

$$\begin{aligned} & \underset{\mathbf{b}, \hat{\mathbf{b}}}{\text{maximize}} \quad \begin{cases} - \sum_{n,m=1}^N (\hat{b}_n - b_n)(\hat{b}_m - b_m) \kappa_{\mathbb{C}}^r(\mathbf{z}_m, \mathbf{z}_n) \\ - \epsilon \sum_{n=1}^N (\hat{b}_n + b_n) + \sum_{n=1}^N d_n^i (\hat{b}_n - b_n) \end{cases} \\ & \text{subject to} \quad \sum_{n=1}^N (\hat{b}_n - b_n) = 0 \text{ and } b_n, \hat{b}_n \in [0, C/N]. \end{aligned} \quad (22b)$$

Observe that (22a) and (22b), are equivalent with the dual problem of a standard real support vector regression task with kernel $2\kappa_{\mathbb{C}}^r$. This is a real kernel, as Lemma 2 establishes. Therefore (figure 2), one may solve the two real SVR tasks for a_n, \hat{a}_n and c^r, b_n, \hat{b}_n, c^i respectively, using any one of the algorithms, which have been developed for this purpose, and then combine the two solutions to find the final non-linear solution of the complex problem as

$$\begin{aligned} g(\mathbf{z}) &= \langle \Phi_{\mathbb{C}}(\mathbf{z}), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}^*(\mathbf{z}), v \rangle_{\mathbb{H}} + c \\ &= 2 \sum_{n=1}^N (\hat{a}_n - a_n) \kappa_{\mathbb{C}}^r(\mathbf{z}_n, \mathbf{z}) + 2i \sum_{n=1}^N (\hat{b}_n - b_n) \kappa_{\mathbb{C}}^r(\mathbf{z}_n, \mathbf{z}) + c. \end{aligned} \quad (23)$$

In this paper we are focusing mainly in the complex Gaussian kernel. It is important to emphasize that, in this case, the induced kernel $\kappa_{\mathbb{C}}^r$ is not the real Gaussian RBF. Figure 1 shows the element $\kappa_{\mathbb{C}}^r(\cdot, (0, 0)^T)$ of the induced real feature space.

Remark 3 For the complexification procedure, we select a real kernel $\kappa_{\mathbb{R}}$ and transform the input data from \mathcal{X} to the complexified space \mathbb{H} , via the feature map $\bar{\Phi}_{\mathbb{C}}$, to obtain the data $\{(\bar{\Phi}_{\mathbb{C}}(\mathbf{z}_n), d_n); n = 1, \dots, N\}$. Following a similar procedure as the one described above and considering that

$$\langle \bar{\Phi}_{\mathbb{C}}(\mathbf{z}_n), \bar{\Phi}_{\mathbb{C}}(\mathbf{z}_m) \rangle_{\mathbb{H}} = 2\kappa_{\mathbb{R}}(\mathbf{z}_m, \mathbf{z}_n)$$

we can easily deduce that the dual of the complexified SVR task is equivalent to two real SVR tasks employing the kernel $2\kappa_{\mathbb{R}}$. Hence, the complexification technique is identical to the DRC approach.

5. Complex Support Vector Machines

Recall that in any real Hilbert space \mathcal{H} , a hyperplane consists of all the elements $f \in \mathcal{H}$ that satisfy

$$\langle f, w \rangle_{\mathcal{H}} + b = 0, \quad (24)$$

for some $w \in \mathcal{H}$, $b \in \mathbb{R}$. Moreover, as figure 3 shows, any hyperplane of \mathcal{H} divides the space into two parts, $\mathcal{H}_+ = \{f \in \mathcal{H}; \langle f, w \rangle_{\mathcal{H}} + b > 0\}$ and $\mathcal{H}_- = \{f \in \mathcal{H}; \langle f, w \rangle_{\mathcal{H}} + b < 0\}$. In the traditional SVM classification task, which has been outlined in section 3, the goal is to separate two distinct classes of data by a maximum margin hyperplane, so that one class falls into \mathcal{H}_+ and the other into \mathcal{H}_- (excluding some outliers). In order to be able to generalize the SVM rationale to complex spaces, firstly, we need to determine an appropriate definition for a complex hyperplane. The difficulty is that the set of complex numbers is not an ordered one, and thus one may not assume that a complex version of (24) divides the space into two parts, as \mathcal{H}_+ and \mathcal{H}_- cannot be defined. Instead, we will provide a novel definition of complex hyperplanes that divide the complex space into four parts. This will be our kick off point for deriving the complex SVM rationale, which classifies objects into four (instead of two) classes.

Lemma 4 *The relations*

$$\operatorname{Re}(\langle f, w \rangle_{\mathbb{H}} + c) = 0, \quad (25a)$$

$$\operatorname{Im}(\langle f, w \rangle_{\mathbb{H}} + c) = 0, \quad (25b)$$

for some $w \in \mathbb{H}$, $c \in \mathbb{C}$, where $f \in \mathbb{H}$, represent two orthogonal hyperplanes of the doubled real space, i.e., \mathcal{H}^2 , in general positions.

Proof Observe that

$$\langle f, w \rangle_{\mathbb{H}} = \langle f^r, w^r \rangle_{\mathcal{H}} + \langle f^i, w^i \rangle_{\mathcal{H}} + \mathbf{i}(\langle f^i, w^r \rangle_{\mathcal{H}} - \langle f^r, w^i \rangle_{\mathcal{H}}),$$

where $f = f^r + \mathbf{i}f^i$, $w = w^r + \mathbf{i}w^i$. Hence, we take that

$$\left\langle \begin{pmatrix} f^r \\ f^i \end{pmatrix}, \begin{pmatrix} w^r \\ w^i \end{pmatrix} \right\rangle_{\mathcal{H}^2} + c^r = 0 \quad \text{and} \quad \left\langle \begin{pmatrix} f^r \\ f^i \end{pmatrix}, \begin{pmatrix} -w^i \\ w^r \end{pmatrix} \right\rangle_{\mathcal{H}^2} + c^i = 0,$$

where $c = c^r + \mathbf{i}c^i$. These are two distinct hyperplanes of \mathcal{H}^2 . Moreover, as

$$\begin{pmatrix} -w^i & w^r \end{pmatrix} \begin{pmatrix} w^r \\ w^i \end{pmatrix} = 0,$$

the two hyperplanes are orthogonal. ■

Lemma 5 *The relations*

$$\operatorname{Re}(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c) = 0, \quad (26a)$$

$$\operatorname{Im}(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c) = 0, \quad (26b)$$

for some $w, v \in \mathbb{H}$, $c \in \mathbb{C}$, where $f \in \mathbb{H}$, represent two hyperplanes of the doubled real space, i.e., \mathcal{H}^2 . Depending on the values of w, v , these hyperplanes may be placed arbitrarily on \mathcal{H}^2 .

Proof Following a similar rationale as in the proof of lemma 4, we finally take

$$\left\langle \begin{pmatrix} f^r \\ f^i \end{pmatrix}, \begin{pmatrix} w^r + v^r \\ w^i - v^i \end{pmatrix} \right\rangle_{\mathcal{H}^2} + c^r = 0$$

and

$$\left\langle \begin{pmatrix} f^r \\ f^i \end{pmatrix}, \begin{pmatrix} -(w^i + v^i) \\ w^r - v^r \end{pmatrix} \right\rangle_{\mathcal{H}^2} + c^i = 0,$$

where $f = f^r + \mathbf{i}f^i$, $w = w^r + \mathbf{i}w^i$, $v = v^r + \mathbf{i}v^i$, $c = c^r + \mathbf{i}c^i$. ■

The following definition comes naturally.

Definition 6 Let \mathbb{H} be a complex Hilbert space. We define the complex couple of hyperplanes as the set of all $f \in \mathbb{H}$ that satisfy one of the following relations

$$\operatorname{Re}(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c) = 0, \quad (27a)$$

$$\operatorname{Im}(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c) = 0, \quad (27b)$$

for some $w, v \in \mathbb{H}$, $c \in \mathbb{C}$.

Lemmas 4 and 5 demonstrate the significant difference between complex linear estimation and widely linear estimation functions, which has been, already, pointed out in section 4.2, albeit in a different context. The complex linear case is quite restrictive, as the couple of complex hyperplanes are always orthogonal. On the other hand, the widely linear case is more general and covers all cases. The complex couple of hyperplanes (as defined by definition 6) divides the space into four parts, i.e.,

$$\begin{aligned} \mathcal{H}_{++} &= \left\{ f \in \mathcal{H}; \begin{array}{l} \operatorname{Re}(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c) > 0, \\ \operatorname{Im}(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c) > 0 \end{array} \right\}, \\ \mathcal{H}_{+-} &= \left\{ f \in \mathcal{H}; \begin{array}{l} \operatorname{Re}(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c) > 0, \\ \operatorname{Im}(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c) < 0 \end{array} \right\}, \\ \mathcal{H}_{-+} &= \left\{ f \in \mathcal{H}; \begin{array}{l} \operatorname{Re}(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c) < 0, \\ \operatorname{Im}(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c) > 0 \end{array} \right\}, \\ \mathcal{H}_{--} &= \left\{ f \in \mathcal{H}; \begin{array}{l} \operatorname{Re}(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c) < 0, \\ \operatorname{Im}(\langle f, w \rangle_{\mathbb{H}} + \langle f^*, v \rangle_{\mathbb{H}} + c) < 0 \end{array} \right\}. \end{aligned}$$

Figure 4 demonstrates a simple case of a complex couple of hyperplanes that divides \mathbb{C} into four parts. Note, that, in some cases, the complex couple of hyperplanes might degenerate into two identical hyperplanes or two parallel hyperplanes.

The complex SVM classification task can be formulated as follows. Suppose we are given training data, which belong to four separate classes $C_{++}, C_{+-}, C_{-+}, C_{--}$, i.e., $\{(z_n, d_n); n = 1, \dots, N\} \subset \mathcal{X} \times \{\pm 1 \pm \mathbf{i}\}$. If $d_n = +1 + \mathbf{i}$, then the n -th sample belongs to C_{++} , i.e., $z_n \in C_{++}$, if $d_n = 1 - \mathbf{i}$, then $z_n \in C_{+-}$, if $d_n = -1 + \mathbf{i}$, then $z_n \in C_{-+}$ and if $d_n = -1 - \mathbf{i}$, then $z_n \in C_{--}$. Consider the complex RKHS \mathbb{H} with respective kernel $\kappa_{\mathbb{C}}$. Following a

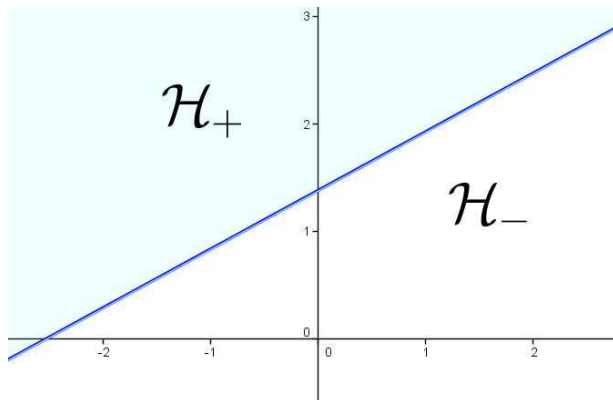


Figure 3: A hyperplane separates the space \mathcal{H} into two parts, \mathcal{H}_+ and \mathcal{H}_- .

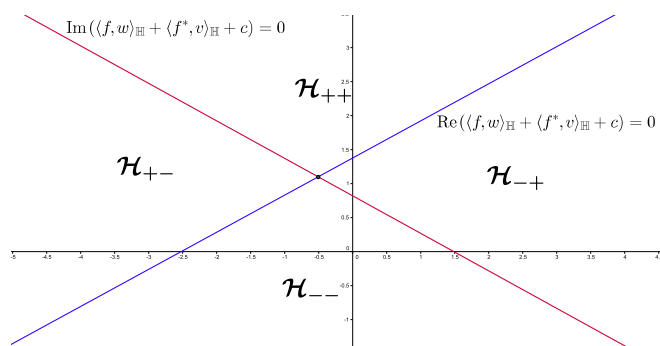


Figure 4: A complex couple of hyperplanes separates the space of complex numbers (i.e., $\mathbb{H} = \mathbb{C}$) into four parts.

similar rationale to the real case, we transform the input data from \mathcal{X} to \mathbb{H} , via the feature map $\Phi_{\mathbb{C}}$. The goal of the SVM task is to estimate a complex couple of maximum margin hyperplanes, that separates the points of the four classes as best as possible (see figure 5). Thus, we need to minimize

$$\begin{aligned} & \left\| \begin{pmatrix} w^r + v^r \\ w^i - v^i \end{pmatrix} \right\|_{\mathcal{H}^2}^2 + \left\| \begin{pmatrix} -(w^i + v^i) \\ w^r - v^r \end{pmatrix} \right\|_{\mathcal{H}^2}^2 = \\ & \|w^r + v^r\|_{\mathcal{H}}^2 + \|w^i - v^i\|_{\mathcal{H}}^2 + \|(w^i + v^i)\|_{\mathcal{H}}^2 + \|w^r - v^r\|_{\mathcal{H}}^2 = \\ & 2\|w^r\|_{\mathcal{H}}^2 + 2\|w^i\|_{\mathcal{H}}^2 + 2\|v^r\|_{\mathcal{H}}^2 + 2\|v^i\|_{\mathcal{H}}^2 = \\ & 2(\|w\|_{\mathbb{H}}^2 + \|v\|_{\mathbb{H}}^2). \end{aligned}$$

Therefore, the primal complex SVM optimization problem can be formulated as

$$\begin{aligned} \min_{w, v, c} \quad & \frac{1}{2}\|w\|_{\mathbb{H}}^2 + \frac{1}{2}\|v\|_{\mathbb{H}}^2 + \frac{C}{N} \sum_{n=1}^N (\xi_n^r + \xi_n^i) \\ \text{s. to} \quad & \begin{cases} d_n^r \operatorname{Re}(\langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}^*(z_n), v \rangle_{\mathbb{H}} + c) \geq 1 - \xi_n^r \\ d_n^i \operatorname{Im}(\langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}^*(z_n), w \rangle_{\mathbb{H}} + c) \geq 1 - \xi_n^i \\ \xi_n^r, \xi_n^i \geq 0 \end{cases} \\ & \text{for } n = 1, \dots, N. \end{aligned} \quad (28)$$

The Lagrangian function becomes

$$\begin{aligned} L(w, v, \mathbf{a}, \hat{\mathbf{a}}, \mathbf{b}, \hat{\mathbf{b}}) = & \frac{1}{2}\|w\|_{\mathbb{H}}^2 + \frac{1}{2}\|v\|_{\mathbb{H}}^2 + \frac{C}{N} \sum_{n=1}^N (\xi_n^r + \xi_n^i) \\ & - \sum_{n=1}^N a_n (d_n^r \operatorname{Re}(\langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}^*(z_n), v \rangle_{\mathbb{H}} + c) - 1 + \xi_n^r) \\ & - \sum_{n=1}^N b_n (d_n^i \operatorname{Im}(\langle \Phi_{\mathbb{C}}(z_n), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}^*(z_n), w \rangle_{\mathbb{H}} + c) - 1 + \xi_n^i) \\ & - \sum_{n=1}^N \eta_n \xi_n^r - \sum_{n=1}^N \theta_n \xi_n^i, \end{aligned}$$

where $a_n, b_n, \eta_n, \theta_n$ are the positive Lagrange multipliers of the respective inequalities, for $n = 1, \dots, N$. To exploit the saddle point conditions of the Lagrangian function, we employ the rules of Wirtinger's Calculus to compute the respective gradients. Hence, we take

$$\begin{aligned} \frac{\partial L}{\partial w^*} &= \frac{1}{2}w - \frac{1}{2} \sum_{n=1}^N a_n d_n^r \Phi_{\mathbb{C}}(z_n) + \frac{\mathbf{i}}{2} \sum_{n=1}^N b_n d_n^i \Phi_{\mathbb{C}}(z_n) \\ \frac{\partial L}{\partial v^*} &= \frac{1}{2}v - \frac{1}{2} \sum_{n=1}^N a_n d_n^r \Phi_{\mathbb{C}}^*(z_n) + \frac{\mathbf{i}}{2} \sum_{n=1}^N b_n d_n^i \Phi_{\mathbb{C}}^*(z_n) \\ \frac{\partial L}{\partial c^*} &= w - \frac{1}{2} \sum_{n=1}^N a_n d_n^r + \frac{\mathbf{i}}{2} \sum_{n=1}^N b_n d_n^i \end{aligned}$$

$$\text{and} \quad \frac{\partial L}{\xi_n^r} = \frac{C}{N} - a_n - \eta_n, \quad \frac{\partial L}{\xi_n^i} = \frac{C}{N} - b_n - \theta_n.$$

for $n = 1, \dots, N$. As all the gradients have to vanish, we finally take that

$$w = \sum_{n=1}^N (a_n d_n^r - \mathbf{i} b_n d_n^i) \Phi_{\mathbb{C}}(\mathbf{z}_n), \quad v = \sum_{n=1}^N (a_n d_n^r - \mathbf{i} b_n d_n^i) \Phi_{\mathbb{C}}^*(\mathbf{z}_n), \quad \sum_{n=1}^N a_n d_n^r = \sum_{n=1}^N b_n d_n^i = 0$$

$$\text{and} \quad a_n + \eta_n = \frac{C}{N}, \quad b_n + \theta_n = \frac{C}{N}$$

for $n = 1, \dots, N$. Following a similar procedure as in the complex SVR case, it turns out that the dual problem can be split into two separate maximization tasks:

$$\begin{aligned} & \underset{\mathbf{a}}{\text{maximize}} \quad \sum_{n=1}^N a_n - \sum_{n,m=1}^N a_n a_m d_n^r d_m^r \kappa_{\mathbb{C}}^r(\mathbf{z}_m, \mathbf{z}_n) \\ & \text{subject to} \quad \begin{cases} \sum_{n=1}^N a_n d_n^r = 0 \\ 0 \leq a_n \leq \frac{C}{N} \end{cases} \quad \text{for } n = 1, \dots, N \end{aligned} \quad (29a)$$

and

$$\begin{aligned} & \underset{\hat{\mathbf{a}}}{\text{maximize}} \quad \sum_{n=1}^N b_n - \sum_{n,m=1}^N b_n b_m d_n^i d_m^i \kappa_{\mathbb{C}}^r(\mathbf{z}_m, \mathbf{z}_n) \\ & \text{subject to} \quad \begin{cases} \sum_{n=1}^N b_n d_n^i = 0 \\ 0 \leq b_n \leq \frac{C}{N} \end{cases} \quad \text{for } n = 1, \dots, N. \end{aligned} \quad (29b)$$

Observe that, similar to the regression case, these problems are equivalent with two distinct real SVM (dual) tasks employing the induced real kernel $2\kappa_{\mathbb{C}}^r$. One may split the (output) data to their real and imaginary parts, as figure 6 demonstrates, solve two real SVM tasks employing any one of the standard algorithms and, finally, combine the solutions to take the complex labeling function:

$$\begin{aligned} g(\mathbf{z}) &= \underset{\mathbf{i}}{\text{sign}}(\langle \Phi_{\mathbb{C}}(\mathbf{z}), w \rangle_{\mathbb{H}} + \langle \Phi_{\mathbb{C}}^*(\mathbf{z}), v \rangle_{\mathbb{H}} + c) \\ &= \underset{\mathbf{i}}{\text{sign}} \left(2 \sum_{n=1}^N (a_n d_n^r + \mathbf{i} b_n d_n^i) \kappa_{\mathbb{C}}^r(\mathbf{z}_n, \mathbf{z}) + c^r + \mathbf{i} c^i \right), \end{aligned}$$

$$\text{where} \quad \underset{\mathbf{i}}{\text{sign}}(z) = \text{sign}(\text{Re}(z)) + \mathbf{i} \text{sign}(\text{Im}(z)).$$

Remark 7 *Following the complexification procedure, as in Remark 3, we select a real kernel $\kappa_{\mathbb{R}}$ and transform the input data from \mathcal{X} to the complexified space \mathbb{H} , via the feature map $\Phi_{\mathbb{C}}$. We can easily deduce that the dual of the complexified SVM task is equivalent to two real SVM tasks employing the kernel $2\kappa_{\mathbb{R}}$.*

Remark 8 *It is evident that both the complex and the complexified SVM can be employed for binary classification as well. The advantage in this case is that one is able to handle complex input data in both scenarios. Moreover, the popular 1-versus-1 and 1-versus-all strategies, which address multiclassification problems, can be directly applied to complex inputs using either the complex or the complexified binary SVM.*

6. Experiments

In order to illuminate the advantages that are gained, if one exploits complex data and to demonstrate the performance of the proposed algorithmic scheme, we compare it with standard real-valued techniques, as well as the dual real channel approach using various regression and classification tasks. In the following, we will refer to the pure complex kernel rationale and the complexification trick, presented in this paper, using the terms CSV (or CSVM) and complexified SV (or complexified SVM) respectively. The dual real channel approach, outlined in section 4.1, will be denoted as DRC-SV. Recall that the DRC approach is equivalent to the complexified rationale, although the latter often provides for more compact formulas and simpler representations. The following experiments were implemented in Matlab. The respective code can be found in bouboulis.mysch.gr/kernels.html.

6.1 Function Estimation

In this section, we perform a simple regression test on the complex function $\text{sinc}(z)$. An orthogonal grid of 33×9 actual points of the *sinc* function, corrupted by a mixture of white Gaussian noise together with some impulses, was adopted as the training data. Figures 7 and 8 show the real and imaginary parts of the reconstructed function using the CSV rationale. Note the excellent visual results obtained by the corrupted training data. Figures 9 and 10 compare the square errors (i.e. $|\hat{d}_n - \text{sinc}(z_n)|^2$, where \hat{d}_n is the value of the estimated function at z_n) between the CSV and the DRC-SV. In this experiment, it is evident that the DRC-SV fails to capture the complex structure of the function. On the other hand, the CSV rationale provides for an estimation function, which exhibits excellent characteristics. A closer look reveals that at the border of the training grid the square error increases in some cases. This is expected, as the available information, which it is exploited by the SV algorithm, is reduced in these areas, compared to the interior points of the grid, making the algorithm more sensitive to outliers. Besides the significant decrease in the square error, in these experiments we, also, observed a significant reduction in the computing time needed for the CSV, compared to the DRC-SV. Both algorithms were implemented in MatLab on a computer with a Core i5 650 microprocessor running at 3.2 GHz. The total computing time for the CSV and the DRC-SV tasks were around 130 and 550 seconds respectively.

In all the performed experiments, the SMO algorithm was employed using the complex Gaussian kernel and the real Gaussian kernel for the CSV and the DRC-SV respectively

(see Platt (1998)). The parameters of the kernel for both the complex SVR and the DRC SVR tasks were tuned to provide the smallest mean square error. In particular for the CSVr, the parameter of the complex gaussian kernel was set to $t = 0.25$, while for the DRC-SVR the parameter was set to $t = 4$. In both cases the parameters of the SVR task were set as $C = 1000$, $\epsilon = 0.1$.

6.2 Channel Identification

In this section, we consider a non-linear channel identification task (see Adali and Li (2010)). This channel consists of the 5-tap linear component:

$$t(n) = \sum_{k=1}^5 h(k) \cdot s(n - k + 1), \quad (30)$$

where $h(k) = 0.432 \left(1 + \cos \left(\frac{2\pi(k-3)}{5} \right) - \left(1 + \cos \frac{2\pi(k-3)}{10} \right) i \right)$, for $k = 1, \dots, 5$, and the nonlinear component:

$$x(n) = t(n) + (0.15 - 0.1i)t^2(n).$$

This is a standard model that has been extensively used in the literature for such tasks, e.g., Sebald and Bucklew (2000); Sanchez-Fernandez et al. (2004); Bouboulis and Theodoridis (2011); Bouboulis et al. (2012b,a). At the receiver's end, the signal is corrupted by white gaussian noise and then observed as y_n . The level of the noise was set to 15dB. The input signal that was fed to the channel had the form

$$s(n) = \left(\sqrt{1 - \rho^2} X(n) + i\rho Y(n) \right), \quad (31)$$

where $X(n)$ and $Y(n)$ are gaussian random variables. This input is circular for $\rho = \sqrt{2}/2$ and highly non-circular if ρ approaches 0 or 1 Adali and Li (2010). The CSVr and the DRC-SVR rationales were used to address the channel identification task, which aims to discover the input-output relationship between $(s(n - L + 1), s(n - L + 2), \dots, s(n))$ and $y(n)$ (the parameter L was set to $L = 5$). In each experiment, a set of 150 pairs of samples was used to perform the training. After training, a set of 600 pairs of samples was used to test the estimation's performance of both algorithms (i.e., to measure the mean square error between the actual channel output, $x(n)$, and the estimated output, $\hat{x}(n)$). To find the best possible values of the parameters C and t , that minimize the mean square error for both SVR tasks, an extensive cross-validation procedure has been employed (see tables 1, 2) in a total of 20 sets of data. Figure 11 shows the minimum mean square error, which has been obtained for all values of the kernel parameter t , versus the SVR parameter C , for both cases. It is evident, that the CSVr approach significantly outperforms the DRC-SVR rationale, both in terms of MSE and computational time (figure 12). Both figures 11 and 12 refer to the circular case. As the results for the non-circular case are similar, they are omitted to save space.

6.3 Channel Equalization

In this section, we present a non-linear channel equalization task that consists of the linear filter (30) and the memoryless nonlinearity $x(n) = t(n) + (0.1 - 0.15i) \cdot t^2(n)$. At the receiver

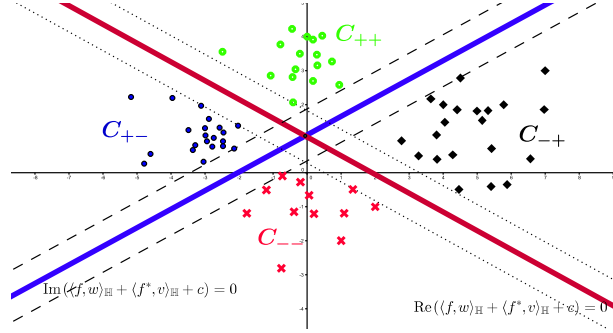


Figure 5: A complex couple of hyperplanes that separates the four given classes. The hyperplanes are chosen so that to maximize the margin between the classes.

| C | t |
|-------|----------|
| 1000 | $1/6^2$ |
| 2000 | $1/6^2$ |
| 5000 | $1/8^2$ |
| 10000 | $1/9^2$ |
| 20000 | $1/11^2$ |
| 50000 | $1/13^2$ |

Table 1: The values of C and t that minimize the mean square error of the CSVN, for the channel identification task.

| C | t |
|-------|----------|
| 1000 | $1/4^2$ |
| 2000 | $1/5^2$ |
| 5000 | $1/6^2$ |
| 10000 | $1/7^2$ |
| 20000 | $1/7^2$ |
| 50000 | $1/10^2$ |

Table 2: The values of C and t that minimize the mean square error of the DRC-SVR, for the channel identification task.

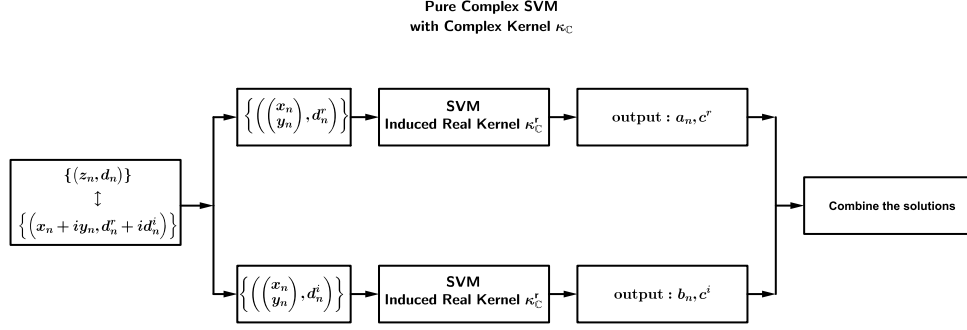


Figure 6: Pure Complex Support Vector Machines.

end of the channel, the signal is corrupted by white Gaussian noise and then observed as $y(n)$. The level of the noise was set to 15dB. The input signal that was fed to the channels had the form

$$s(n) = 0.30 \left(\sqrt{1 - \rho^2} X(n) + i \rho Y(n) \right), \quad (32)$$

where $X(n)$ and $Y(n)$ are gaussian random variables.

The aim of a channel equalization task is to construct an inverse filter, which acts on the output $y(n)$ and reproduces the original input signal as close as possible. To this end, we apply the CSVr and DRC-SVR algorithms to a set of samples of the form

$$((y(n + D), y(n + D - 1), \dots, r(y + D - L + 1)), s(n)),$$

where $L > 0$ is the filter length and D the equalization time delay (in this experiment we set $L = 5$ and $D = 2$).

Similar to the channel identification case, in each experiment, a set of 150 pairs of samples was used to perform the training. After training, a set of 600 pairs of samples was used to test the performance of both algorithms (i.e., to measure the mean square error between the actual input, $s(n)$, and the estimated input, $\hat{s}(n)$). To find the best possible values of the parameters C and t , that minimize the mean square error for both SVr tasks, an extensive cross-validation procedure has been employed (see tables 3, 4) in a total of 100 sets of data. Figure 13 shows the minimum mean square error, which has been obtained for all values of the kernel parameter t , versus the SVr parameter C , for both cases considering a circular input. The CSVr appears to achieve a slightly lower MSE for all values of the parameter C , at the cost of a slightly increased computational time. The results for the non-circular case are similar.

6.4 One versus three multiclass Classification

We conclude the experimental section with the classification case. We performed two experiments using the popular MNIST database of handwritten digits LeCun and Cortes. In both cases, the respective parameters of the SVM tasks were tuned to obtain the lowest error rate possible. The MNIST database contains 60000 handwritten digits (from 0 to 9)

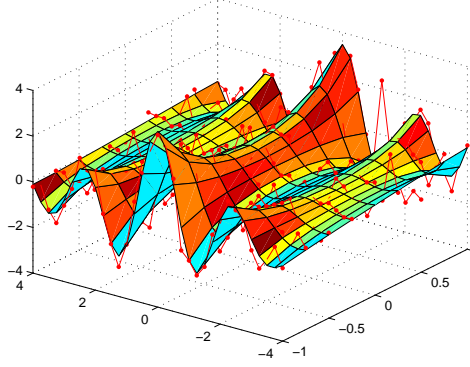


Figure 7: The real part ($\text{Re}(\text{sinc}(z))$) of the estimated *sinc* function from the complex SVR. The points shown in the figure are the real parts of the noisy training data used in the simulation.

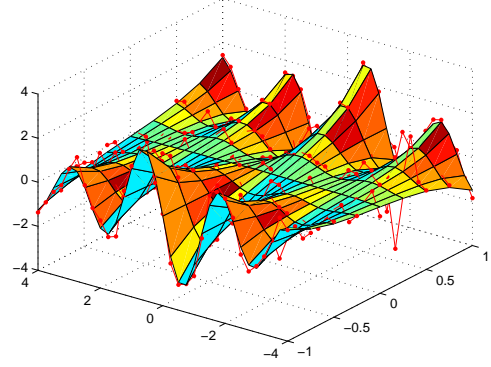


Figure 8: The imaginary part ($\text{Im}(\text{sinc}(z))$) of the estimated *sinc* function from the complex SVR. The points shown in the figure are the imaginary parts of the noisy training data used in the simulation.

| C | t |
|------|-----------|
| 1 | $1/2.5^2$ |
| 2 | $1/2.5^2$ |
| 5 | $1/2.5^2$ |
| 10 | $1/3^2$ |
| 50 | $1/4.5^2$ |
| 100 | $1/5.5^2$ |
| 200 | $1/6^2$ |
| 500 | $1/7^2$ |
| 1000 | $1/9^2$ |

Table 3: The values of C and t that minimize the mean square error of the CSVN, for the channel equalization task.

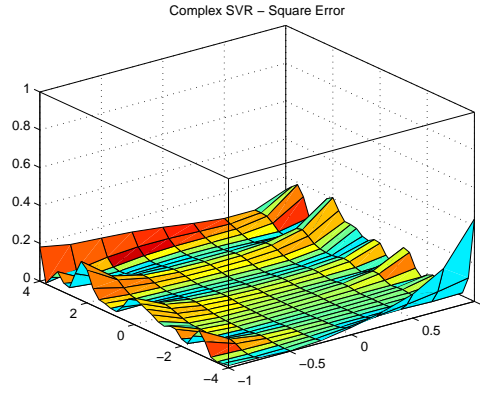


Figure 9: The square error (the actual values of the function versus the estimated ones) for the complex SVR of the sinc function. The mean square error of all the estimated values was equal to $-14.5dB$.

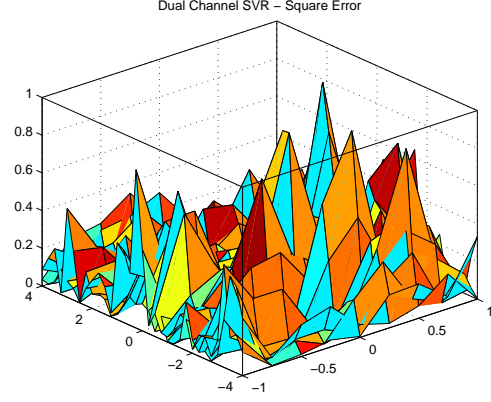


Figure 10: The square error (the actual values of the function versus the estimated ones) for the Dual Real Channel regression of the sinc function. The mean square error of all the estimated values was equal to $-9.5dB$.

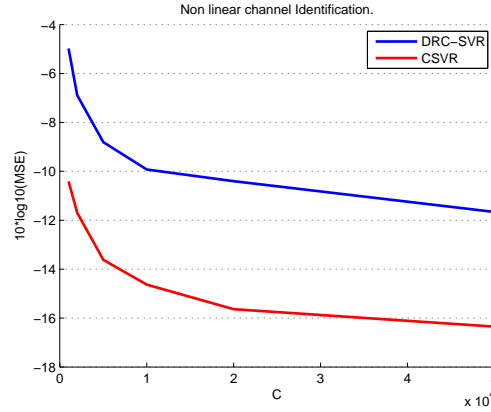


Figure 11: MSE versus the SVR parameter C for both the CSVR and the DRC-SVR rationales, for the channel identification task.

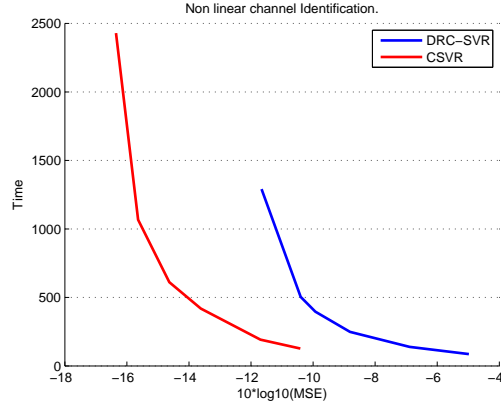


Figure 12: Time (in seconds) versus MSE (dB) for both the CSV and the DRC-SVR rationales, for the channel identification task.

| C | t |
|------|------------|
| 1 | $1/1.5^2$ |
| 2 | $1/1.75^2$ |
| 5 | $1/1.75^2$ |
| 10 | $1/2.25^2$ |
| 50 | $1/2.5^2$ |
| 100 | $1/3^2$ |
| 200 | $1/5^2$ |
| 500 | $1/7^2$ |
| 1000 | $1/7.5^2$ |

Table 4: The values of C and t that minimize the mean square error of the DRC-SVR, for the channel equalization task.

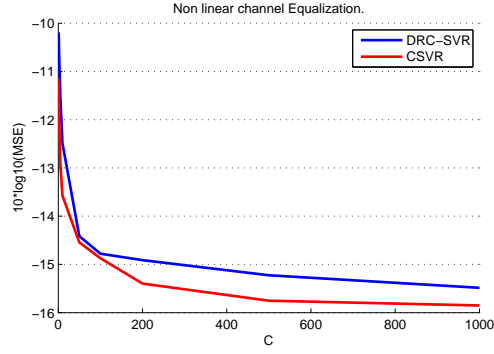


Figure 13: MSE versus the SVR parameter C for both the CSV and the DRC-SVR rationales, for the channel equalization task.

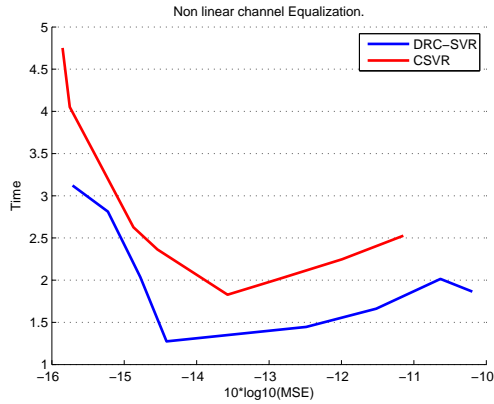


Figure 14: Time (in seconds) versus MSE (dB) for both the CSV and the DRC-SVR rationales, for the channel equalization task.

for training and 10000 handwritten digits for testing. Each digit is encoded as an image file with 28×28 pixels. The scenario, that it is typically used to quantify the performance of an SVM-like learning machine, is to employ a one-versus-all strategy to the training set (using the raw pixel values as input data) and then measure the success using the testing set LeCun et al. (1998); Decoste and with (2002).

In the first experiment, we compare the aforementioned standard one-versus-all scenario with a classification task that exploits complex numbers. In the complex variant, we perform a Fourier transform to each training image and keep only the 100 most significant coefficients. As these coefficients are complex numbers, we employ a one-versus-all classification task using the binary complexified SVM rationale (see remark 8. In both scenarios we use the first 6000 digits of the MNIST training set to train the learning machines and test their performances using the 10000 digits of the testing set. In addition, we used the gaussian kernel with $t = 1/64$ and $t = 1/140^2$ respectively. The SVM parameter C has been set equal to 100. The error rate of the standard real-valued scenario is 3.79%, while the error rate of the complexified (one-versus-all) SVM is 3.46%. In both learning tasks we used the SMO algorithm to train the SVM. The total amount of time needed to perform the training of each learning machine is almost the same for both cases (the complexified task is slightly faster).

In section 5, we discussed how the 4-classes problem comes naturally to the complex SVM. Exploiting the notion of the complex couple of hyperplanes (see figure 4), we have shown that the generalization of the SVM rationale to complex spaces directly assumes quaternary classification. Using this approach, the 4 classes problem can be solved using only 2 distinct SVM tasks instead of the 4 tasks needed by the 1-versus-all or the 1-versus-1 strategies. The second experiment compares the quaternary complex SVM approach to the standard 1-versus-all scenario using the first four digits (0, 1, 2 and 3). In both cases we used the first 6000 such digits of the MNIST training set to train the learning machines. We tested their performance using the digits contained in the testing set. The error rate of the 1-versus-all SVM was 0.721%, while the error rate of the complexified SVM was 0.866%. However, the 1-versus-all SVM task required about double the time for training, compared to the complexified SVM. This is expected, as the latter solves half as many distinct SVM tasks as the first one. In both experiments we used the gaussian kernel with $t = 1/49$ and $t = 1/160^2$ respectively. The SVM parameter C has been set equal to 100 in this case also.

7. Conclusions

We presented a framework of support vector regression and quaternary classification for complex data using pure complex kernels, or complexified real ones, exploiting the recently developed Wirtinger's calculus for complex RKHS's and the notions of widely linear estimation. We showed that this problem is equivalent to solving two separate real SVM tasks employing an induced real kernel (figure 2). The induced kernel depends on the choice of the complex kernel and it is not one of the usual kernels used in the literature. Although the machinery presented here might seem similar to the dual channel approach, they have important differences. The most important one is due to the incorporation of the induced kernel κ_1 , which allows us to exploit the complex structure of the space, which is lost in the dual channel approach. As an example we studied the complex Gaussian kernel and

showed by example that the induced kernel is not the real Gaussian RBF. To the best of our knowledge this kernel has not appeared before in the literature. Hence, treating complex tasks directly in the complex plane, opens the way of employing novel kernels.

Furthermore, for the classification problem we have shown that the complex SVM solves directly a quaternary problem, instead of the binary problem, that it is associated to the real SVM. Hence, the complex SVM not only provides the means for treating complex inputs, but also offers an alternative strategy to address multiclassification problems. In this way, such problems can be solved faster (needing about the half time), at a cost of increased error rate (in our experiment the increase was about 19%). Although, in the present work we focused on the 4 classes problem only, it is evident that the same rationale can be carried out to any multidimensional problem, were the classes must be divided into four groups each time, following a rationale similar to the one-versus-all mechanism. This will be addressed at a future time.

Acknowledgments

This work was carried out under the 621 ARISTEIA program, co-financed by the Greek Secretariat for Research and Development and the EU.

References

- T. Adali and H. Li. *Adaptive signal processing: next generation solutions*, chapter Complex-valued Adaptive Signal Processing, pages 1–74. Wiley, NJ, 2010.
- N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, May 1950.
- F. Bach. Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research*, 9:1179–1225, 2008.
- F. Bach, G. Lanckriet, and M. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *21st International Conference on Machine Learning*, pages 41–48, 29 2010-sept. 1 2004.
- E.J. Bayro-Corrochano and N. Arana-Daniel. Clifford support vector machines for classification, regression, and recurrence. *Neural Networks, IEEE Transactions on*, 21(11):1731–1746, nov. 2010. ISSN 1045-9227. doi: 10.1109/TNN.2010.2060352.
- P. Bouboulis and M. Mavroforakis. Reproducing kernel Hilbert spaces and fractal interpolation. *Journal of Computational and Applied Mathematics*, 235(12):3425–3434, April 2011. doi: 10.1016/j.cam.2011.02.003.
- P. Bouboulis and S. Theodoridis. Extension of Wirtinger’s calculus to reproducing kernel Hilbert spaces and the complex kernel LMS. *IEEE Transactions on Signal Processing*, 59(3):964–978, 2011.

- P. Bouboulis, K. Slavakis, and S. Theodoridis. Adaptive kernel-based image denoising employing semi-parametric regularization. *IEEE Transactions on Image Processing*, 19(6):1465–1479, 2010.
- P. Bouboulis, K. Slavakis, and S. Theodoridis. Adaptive learning in complex reproducing kernel Hilbert spaces employing Wirtinger’s subgradients. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(3):425–438, march 2012a. ISSN 2162-237X. doi: 10.1109/TNNLS.2011.2179810.
- P. Bouboulis, S. Theodoridis, and M. Mavroforakis. The augmented complex kernel LMS. *Signal Processing, IEEE Transactions on*, 60(9):4962–4967, 2012b. ISSN 1053-587X. doi: 10.1109/TSP.2012.2200479.
- A.S. Cacciapuoti, G. Gelli, L. Paura, and F. Verde. Finite-sample performance analysis of widely linear multiuser receivers for DS-CDMA systems. *IEEE Transactions on Signal Processing*, 56(4):1572 – 1588, 2008.
- B. Che Ujang, C.C. Took, and D.P. Mandic. Quaternion-valued nonlinear adaptive filtering. *Neural Networks, IEEE Transactions on*, 22(8):1193–1206, aug. 2011. ISSN 1045-9227. doi: 10.1109/TNN.2011.2157358.
- P. Chevalier and F. Pipon. New insights into optimal widely linear receivers for the demodulation of BPSK, MSK and GMSK signals corrupted by noncircular interferences - application to SAIC. *IEEE Transactions on Signal Processing*, 54(3):870–883, 2006.
- R. Collobert and S. Bengio. SVMtorch: Support vector machines for large-scale regression problems. *Journal of Machine Learning Research*, 1, 2001.
- D. Decoste and B. Predicting Time Series with. Training invariant support vector machines. *Machine Learning*, 46, 161190, 2002, 46:161–190, 2002.
- Y. Engel, S. Mannor, and R. Meir. The kernel recursive least-squares algorithm. *IEEE Trans. Signal Process.*, 52(8):2275–2285, 2004.
- C.C. Gaudes, I. Santamaria, J. Via, E.M.M. Gomez, and T.S. Paules. Robust array beamforming with sidelobe control using support vector machines. *Signal Processing, IEEE Transactions on*, 55(2):574–584, feb. 2007. ISSN 1053-587X. doi: 10.1109/TSP.2006.885720.
- M. Gönen and E. Alpaydin. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011.
- J. J. Jeon, J. G. Andrews, and K. M. Sung. The blind widely linear output energy algorithm for DS-CDMA systems. *IEEE Transactions on Signal Processing*, 54(5):1926–1931, 2006.
- J. Kivinen, A. Smola, and R. C. Williamson. Online learning with kernels. *IEEE Trans. Signal Process.*, 52(8):2165–2176, 2004.
- A. Kuh and D. P. Mandic. Applications of complex augmented kernels to wind profile prediction. *Proceedings of ICASSP*, 2009.

- Jorge López Lázaro. *Analysis and Convergence of SMO-like Decomposition and Geometrical Algorithms for Support Vector Machines*. PhD thesis, Universidad Autónoma de Madrid, November 2011.
- Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Yann LeCun and Corinna Cortes. The MNIST database. URL <http://yann.lecun.com/exdb/mnist/>.
- H. Li. *Complex-valued adaptive signal processing using Wirtinger calculus and its application to Independent Component Analysis*. PhD thesis, University of Maryland Baltimore County, 2008.
- W. Liu, P. Pokharel, and J. C. Principe. The kernel Least-Mean-Square algorithm. *IEEE Trans. Signal Process.*, 56(2):543–554, 2008.
- W. Liu, J. C. Principe, and S. Haykin. *Kernel Adaptive Filtering*. Wiley, 2010.
- D. Mandic and V.S.L Goh. *Complex Valued nonlinear Adaptive Filters*. Wiley, 2009.
- M.E. Mavroforakis and S. Theodoridis. A geometric approach to support vector machine (SVM) classification. *IEEE Transactions on Neural Networks*, 17(3):671–682, May 2006.
- M.E. Mavroforakis, M. Sdralis, and S. Theodoridis. A geometric nearest point algorithm for the efficient solution of the SVM classification task. *IEEE Transactions on Neural Networks*, 18(5):1545–1549, 2007. doi: 10.1109/TNN.2007.900237.
- K. R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. *Proceedings MSRS*, 1997.
- K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning. *IEEE Trans. Neural Networks*, 12(2):181–201, 2001.
- M. Novey and T. Adali. On extending the complex ICA algorithm to noncircular sources. *IEEE Transactions on Signal Processing*, 56(5):2148–2154, 2008.
- V. I. Paulsen. An Introduction to the theory of Reproducing Kernel Hilbert Spaces. Notes, September 2009. URL www.math.uh.edu/~vern/rkhs.pdf.
- B. Picinbono. On circularity. *IEEE Transactions on Signal Processing*, 42(12):3473–3482, 1994.
- B. Picinbono and P. Bondon. Second order statistics of complex signals. *IEEE Trans. Signal Process.*, 45(2):411–420, 1997.
- B. Picinbono and P. Chevalier. Widely linear estimation with complex data. *IEEE Transactions on Signal Processing*, 43(8):2030–2033, 1995.
- J. Platt. Sequential minimal optimization: A fast algorithm for training Support Vector Machines. Technical Report MSR-TR-98-14, Microsoft Research, April 1998.

- M.M. Ramon, Nan Xu, and C.G. Christodoulou. Beamforming using support vector machines. *Antennas and Wireless Propagation Letters, IEEE*, 4:439 – 442, 2005. ISSN 1536-1225. doi: 10.1109/LAWP.2005.860196.
- M. Sanchez-Fernandez, M. de Prado-Cumplido, J. Arenas-Garcia, and F. Perez-Cruz. Svm multiregression for nonlinear channel estimation in multiple-input multiple-output systems. *Signal Processing, IEEE Transactions on*, 52(8):2298 – 2307, aug. 2004. ISSN 1053-587X. doi: 10.1109/TSP.2004.831028.
- Bernhard Scholkopf, Koji Tsuda, and Jean-Philippe Vert. *Kernel methods in computational biology*. MIT Press, 2004.
- B. Scholkopf and A.J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.
- D.J. Sebald and J.A. Bucklew. Support vector machine techniques for nonlinear equalization. *Signal Processing, IEEE Transactions on*, 48(11):3217 – 3226, nov 2000. ISSN 1053-587X. doi: 10.1109/78.875477.
- J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
- A. Shilton and D.T.H. Lai. Quaternionic and complex-valued support vector regression for equalization and function approximation. In *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, pages 920 – 925, aug. 2007. doi: 10.1109/IJCNN.2007.4371081.
- A. Shilton, D.T.H. Lai, and M. Palaniswami. A division algebraic framework for multidimensional support vector regression. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 40(2):517 – 528, april 2010. ISSN 1083-4419. doi: 10.1109/TSMCB.2009.2028314.
- A. Shilton, D.T.H. Lai, B.K. Santhiranayagam, and M. Palaniswami. A note on octonionic support vector regression. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(3):950 – 955, june 2012. ISSN 1083-4419. doi: 10.1109/TSMCB.2011.2170564.
- K. Slavakis, S. Theodoridis, and I. Yamada. On line kernel-based classification using adaptive projection algorithms. *IEEE Transactions on Signal Processing*, 56(7):2781–2796, 2008.
- K. Slavakis, S. Theodoridis, and I. Yamada. Adaptive constrained learning in reproducing kernel Hilbert spaces: the robust beamforming case. *IEEE Transactions on Signal Processing*, 57(12):4744–4764, 2009.
- K. Slavakis, P. Bouboulis, and S. Theodoridis. Online learning in reproducing kernel Hilbert spaces. *Elsevier’s E-Reference Signal Processing*, 1, section 5, article number 33:1–76, 2013 (to appear).

- A. Smola and B. Schölkopf. A tutorial on support vector regression. Technical Report NC-TR-98-030, Royal Holloway Coll., Univ. London, London, UK Tech, 1998.
- S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.
- S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 4th edition, Nov. 2008.
- S. Theodoridis and M.E. Mavroforakis. Reduced convex hulls: A geometric approach to Support Vector Machines. *IEEE Signal Processing Magazine*, 24(3):119–122, May 2007.
- C. Cheong Took and D. P. Mandic. A quaternion widely linear adaptive filter. *IEEE Transactions on Signal Processing*, 58(8):4427–4431, 2010.
- J. Vanderbei. Loqo: an interior point code for quadratic programming. Technical report, Statistics and Operations Research, Princeton University, NJ, 1994.
- V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1999.
- W. Wirtinger. Zur formalen theorie der functionen von mehr complexen veranderlichen. *Mathematische Annalen*, 97:357–375, 1927.